CLAIMS

What is claimed is:

1. A method comprising:
receiving an interrupt;
converting said interrupt into an upstream memory write interrupt; and
converting said upstream memory write interrupt into a front side bus (FSB) interrupt transaction.

2. The method as in claim 1, wherein said interrupt is generated by a peripheral component interconnect (PCI) device.

3. The method as in claim 2, wherein said FSB interrupt is received by a processor.

4. A method comprising:
receiving a message signaled interrupt (MSI) interrupt;
forwarding said MSI interrupt; and
converting said MSI interrupt into an FSB interrupt transaction.

5. The method as in claim 4, wherein said MSI interrupt is generated by a PCI device, and wherein said FSB interrupt is received by a processor.

6. The method as in claim 5, wherein said FSB interrupt is received by a processor.

7. A method comprising:
receiving a hardware signal;
converting said hardware signal into an upstream memory write interrupt; and
converting said upstream memory write interrupt into an FSB interrupt transaction.

8. The method as in claim 7, wherein said hardware signal is generated by a PCI device, and wherein said FSB interrupt is received by a processor.

9. The method as in claim 8, wherein said FSB interrupt is received by a processor.

10. An apparatus comprising:

a chipset, configured to receive an interrupt and for converting said interrupt into an FSB interrupt transaction.

11. The apparatus as in claim 10, said chipset further comprising at least one of an I/O controller hub (ICH), P64H, AGP device.

12. The apparatus as in claim 11, further comprising an I/O component of an advanced programmable interrupt controller (IOxAPIC) configured to convert said interrupt into an upstream memory write interrupt.

13. The apparatus as in claim 12, said chipset further comprising a HUB interface coupled on a first end to said IOxAPIC and coupled on a second end to a MCH, wherein said memory controller hub (MCH) configured to convert said upstream memory write interrupt into a FSB interrupt transaction.

14. The apparatus as in claim 10, wherein said interrupt is generated by a PCI device, and wherein said chipset is coupled to a processor.

15. The apparatus as in claim 10, wherein said interrupt is a MSI interrupt.

16. The apparatus as in claim 15, said chipset further comprising a HUB interface coupled on a first end to an ICH and coupled on a second end to a MCH, wherein said ICH configured to forward said MSI interrupt to said HUB interface, and wherein said MCH configured to convert said MSI interrupt into a FSB interrupt transaction.

17. The apparatus as in claim 16, wherein said MCH configured to ensure at least one data pipe of a HUB interface is flushed upstream before propagating an interrupt upstream.

18. The apparatus as in claim 16, wherein said MCH configured to receive an end of interrupt (EOI) from a processor and broadcast said EOI to at least one

downstream HUB interface IOxAPIC generating at least one level mode interrupt.

19. An apparatus comprising:

an MCH configured to redirect at least one interrupt based on task priority information.

20. The apparatus as in claim 19, wherein said task priority information is downloaded by a processor into said MCH having a task priority register (TPR) and FSB XTPR update transactions.

21. The apparatus as in claim 19, wherein said interrupt is an upstream memory write interrupt.

22. The apparatus as in claim 19, wherein said interrupt is an IPI interrupt.

23. The apparatus as in claim 19, wherein said interrupt is marked as lowest priority re-directable, and redirected to a lowest priority register.

24. A method comprising

redirecting at least one interrupt based on task priority information.

25. The method as in claim 24, wherein said interrupt is an upstream memory write interrupt.

26. The method as in claim 24, wherein said interrupt is an IPI interrupt.

27. The method as in claim 24, wherein said interrupt is marked as lowest priority re-directable, and redirected to a lowest priority register.

28. A method comprising:

providing preferred ordering of at least one XTPR update transaction and at least one interrupt to be redirected.

29. The method as in claim 28, further comprising ensuring valid information is used for an interrupt redirection when an XTPR update transaction and an interrupt occur about the same time.

30. The method as in claim 29, further comprising providing support for said XTPR update transactions to update at least one XTPR register.

Add A1